CLAIMS IN THE CASE

Please amend Claims 9-11, 13, and 25-26, as follows.

1. (Original) A system for checking consistency of a lock-step process comprising:

a production microcontroller installed on a test circuit, the production microcontroller running a microcontroller code and producing a first value;

an ICE (in circuit emulator) coupled to a host device, wherein the ICE emulates the production microcontroller to form a virtual microcontroller, the virtual microcontroller running the microcontroller code simultaneously with the production microcontroller and producing a second value; and

B4 an interface for coupling the production microcontroller and the ICE enabling transmission of the first value to the ICE, wherein the ICE compares the first value against the second value.

2. (Original) A system for checking consistency of a lock-step process as recited in Claim 1 wherein the microcontroller is installed on a POD.

3. (Original) A system for checking consistency of a lock-step process as recited in Claim 1 wherein the microcontroller is substantially copied in a field programmable gate array (FPGA) in the ICE.

4. (Original) A system for checking consistency of a lock-step process as recited in Claim 1 where the production microcontroller sends to the ICE a result of an execution of a line of code.

5. (Original) A system for checking consistency of a lock-step process as recited in Claim 1 where the ICE ignores execution of the microcontroller code if the microcontroller code executed by the microcontroller is an I/O instruction.

6. (Original) A system for checking consistency of a lock-step process as recited in Claim 1 where the ICE ignores execution of the microcontroller code if there is an interrupt in operation of the microcontroller.

7. (Original) A system for checking consistency of a lock-step process as recited in Claim 1 wherein the ICE stores the second value for comparison with the first value received from the production microcontroller.

8. (Original) A system for checking consistency of a lock-step process as recited in Claim 1 wherein the ICE sends an error signal when the first value and the second value are inconsistent.

13 cont'd

9. (Currently Amended) A system for checking consistency of a lock-step process as recited in Claim 1 wherein the ICE independently ~~detect~~ detects inconsistency in the first value and the second value when there is an interrupt in operation of the production microcontroller.

10. (Currently Amended) A system for checking consistency of a lock-step process as recited in Claim 1 wherein the production microcontroller ~~may receive~~ receives data from a test circuit.

11. (Currently Amended) A system for checking consistency of a lock-step process as recited in Claim 1 wherein the production microcontroller ~~may receive~~ receives data from a product ~~that is intended to be~~ controlled by the microcontroller.

12. (Original) A system for checking consistency of a lock-step process as recited in Claim 1 wherein the production microcontroller provides the ICE with I/O read information sent by the test circuit.

13. (Currently Amended) A system for checking consistency of a lock-step process as recited in Claim 1 wherein the production microcontroller

provides the ICE with I/O read information sent by a product ~~that is intended to be~~ controlled by the microcontroller.

14. (Original) A system for checking consistency of a lock-step process as recited in Claim 1 wherein the ICE compares the first and the second values and where there is inconsistency a software residing in the ICE halts the execution of microcontroller and issues an error signal.

15. (Original) A method for verifying lock step microcontroller code execution in a code debugging process comprising:

a) initializing a first memory of an ICE (in circuit emulator) and a second memory of a microcontroller with microcontroller test code;

b) executing the microcontroller test code on the microcontroller and on the ICE in lock step;

c) computing an ICE current instruction result and a microcontroller current instruction result;

d) verifying lock step execution by comparing the ICE current instruction result and the microcontroller current instruction result;

d) if lock step execution is not verified, halting the execution of the microcontroller test code and reporting an error; and

e) if lock step execution is verified, continuing execution of the microcontroller test code.

16. (Original) The method of Claim 15 further comprising:

if lock step execution is not verified, saving an execution history using a trace buffer coupled to the ICE.

17. (Original) The method of Claim 15 further comprising:

transmitting the microcontroller current instruction result from the microcontroller to the ICE for verifying the lock step execution.

18. (Original) The method of Claim 15 wherein the ICE ignores execution of an instruction if the instruction is an I/O instruction.

19. (Original) The method of Claim 15 wherein the ICE ignores execution of an instruction if the instruction is an interrupt.

20. (Original) The method of Claim 15 wherein the ICE independently detects inconsistency between the ICE current instruction result and the microcontroller current instruction result when the execution of the microcontroller test code is halted.

21. (Original) A lock step code execution microcontroller testing system comprising:

a microcontroller installed on a test circuit, the microcontroller running microcontroller code, the microcontroller is installed on a POD;

an ICE (in circuit emulator) coupled to a host device, wherein the ICE emulates the microcontroller using a field programmable gate array (FPGA) in the ICE, the ICE running the microcontroller code simultaneously with the microcontroller; and

an interface for coupling the microcontroller and the ICE enabling data transmission between the ICE and the microcontroller, wherein the ICE compares a microcontroller execution result with an ICE execution result to verify lock step microcontroller code execution.

22. (Original) The system of Claim 21 wherein the ICE ignores execution of the microcontroller code if the microcontroller code executed by the microcontroller is an I/O instruction.

23. (Original) The system of Claim 21 wherein the ICE ignores execution of the microcontroller code if the microcontroller code is an interrupt instruction.

24. (Original) The system of Claim 21 wherein the ICE independently detects inconsistency between the microcontroller execution result and the ICE execution result when there is a halt in the operation of the microcontroller.

25. (Currently Amended) The system of Claim 21 wherein the microcontroller is coupled to receive data from a product ~~that is intended to be~~ controlled by the microcontroller.

26. (Currently Amended) The system of Claim 21 wherein the microcontroller provides the ICE with I/O read information sent by a product ~~that is intended to be~~ controlled by the microcontroller.